Web Services: Introduction and State of The Art

Óscar Corcho-García, José Carlos del Arco-Prieto and Jesús Arias-Fisteus

In this paper we provide a brief introduction to Web Services, including the main specifications: SOAP, WSDL and UDDI. We also describe other specifications that complement them and provide solutions to aspects needed to develop service oriented architectures based on Web Services. We also address some of the research issues open, including the semantic description of services, which is one of the issues to which more effort is being devoted currently. Finally, we list the main areas where Web Service technology is being applied successfully in the context of enterprises.

Keywords: B2B, BPM, EAI, Integration, Specifications, Web Services.

1 Introduction

In the first generations of the Web, the Web was seen as a collection of information available either in a static way (static Web documents normally generated manually by persons) or in a dynamic way (dynamic Web documents generated from databases, creating the so-called Deep Web).

Web Service technology allowed lifting the Web to a new level of service, where software applications and components could be accessed and executed using the Web as their medium for transmitting inputs needed for the execution and outputs obtained from it. Specifications (and their corresponding components) like UDDI (*Universal Description, Discovery and Integration*), WSDL (*Web Services Description Language*) and SOAP (*Simple Object Access Protocol*) are the basis for this new type of Web. They will be described in Section 3 of this document.

The use of this set of specifications and their corresponding components is tightly related to the development of the *Service Oriented Architecture* (SOA) vision, with the idea

Authors

Oscar Corcho-García works as a Marie Curie fellow at the University of Manchester. Previously, he has worked at iSOCO as a research manager and at the Ontological Engineering Group of the *Universidad Politécnica de Madrid* (UPM). He graduated with honours in Computer Science from UPM in 2000, obtained his MSc in Software Engineering from UPM in 2001, and his PhD in Artificial Intelligence in 2004. He received the third Spanish award in Computer Science from the Spanish Government (2001) and a PhD thesis award from the *Universidad Politécnica de Madrid* (2005). His research activities include the Semantic Grid, the Semantic Web, and ontology engineering. He has participated in several leading EU projects in these areas: OntoGrid, Esperonto, DIP, HOPS, SWWS, Knowledge Web, OntoWeb, and MKBEEM. He has also taken part in the HALO project, funded by Vulcan, Inc. He has published the books "*Ontological Engineering*" and "*A layered declarative approach to ontology translation with knowledge preservation*", over 30 journal and conference/workshop papers, and he reviews papers in several conferences, workshops and journals. He has also been a research visitor at KMI (Open University) and SMI (Stanford University). He chaired the demo/industrial sessions at EKAW2002, co-organized the ISWC2003 and ISWC2004 Workshops on Evaluation of Ontology Tools (EON2003, EON2004) and was the sponsor chair of the ESWC2006 conference. <oscar.corcho@manchester.ac.uk>.

Jose Carlos del Arco-Prieto is a graduate in Computer Science from the *Universidad de Huelva* (1998). Later his professional career began in Tecsidel (1999) where he participated in projects for Telefónica I+D and made analyses of the impact of Web Services on different markets (2002). In 2004 he worked in the *Diputacion de Huelva* (Huelva Provincial Council) participating in the European project "*Regions on Line*" (ROL). In 2005 he worked in Cibernos collaborating on the Diraya project led by Indra and participating in the development of proposals of interoperability models and Web Services-based integration. He is currently working for T-Systems, doing technical consultancy work and collaborating in the definition of interoperability models, integration for the public sector, and the design of methodologies for SOA development. He collaborates with the *Universidad de Huelva* in the promotion of Web 2.0 technologies. He is the father of the first Web Services Latin list (webservices-Latinos), promoter of JSWEB 2005 Workshop, Co-president of JSWEB 2006 Workshop and a member of the steering committee of European SOA & Application Architecture Conference (2007). One of his main interests is the convergence between the worlds of academia and business. <josecarlos.delarco@t-systems.es>.

Jesús Arias-Fisteus works as an assistant professor in the Telematic Engineering Department of the *Universidad Carlos III de Madrid*. He received his MSc with honours in Telecommunication Engineering in 2001 from the *Universidad de Vigo*. In 2005, he received his PhD in Communication Technologies from the *Universidad Carlos III de Madrid*. His research topics include the application of formal methods, especially model checking, to the verification of business processes and Web Service compositions. Recently, he has also become interested in the Semantic Web and Semantic Web Services. He has worked on several European and Spanish research projects related to the field of Web Services. He has also authored more than 15 papers in national and international journals and workshops / conferences in related fields. He was a temporary research visitor at the IT Innovation Centre at Intel Ireland in 2004, and a visiting scientist under the direction of Prof. Tim Berners-Lee at the Decentralized Information Group of the Massachusetts Institute of Technology in 2006. <jaf@it.uc3m.es>.

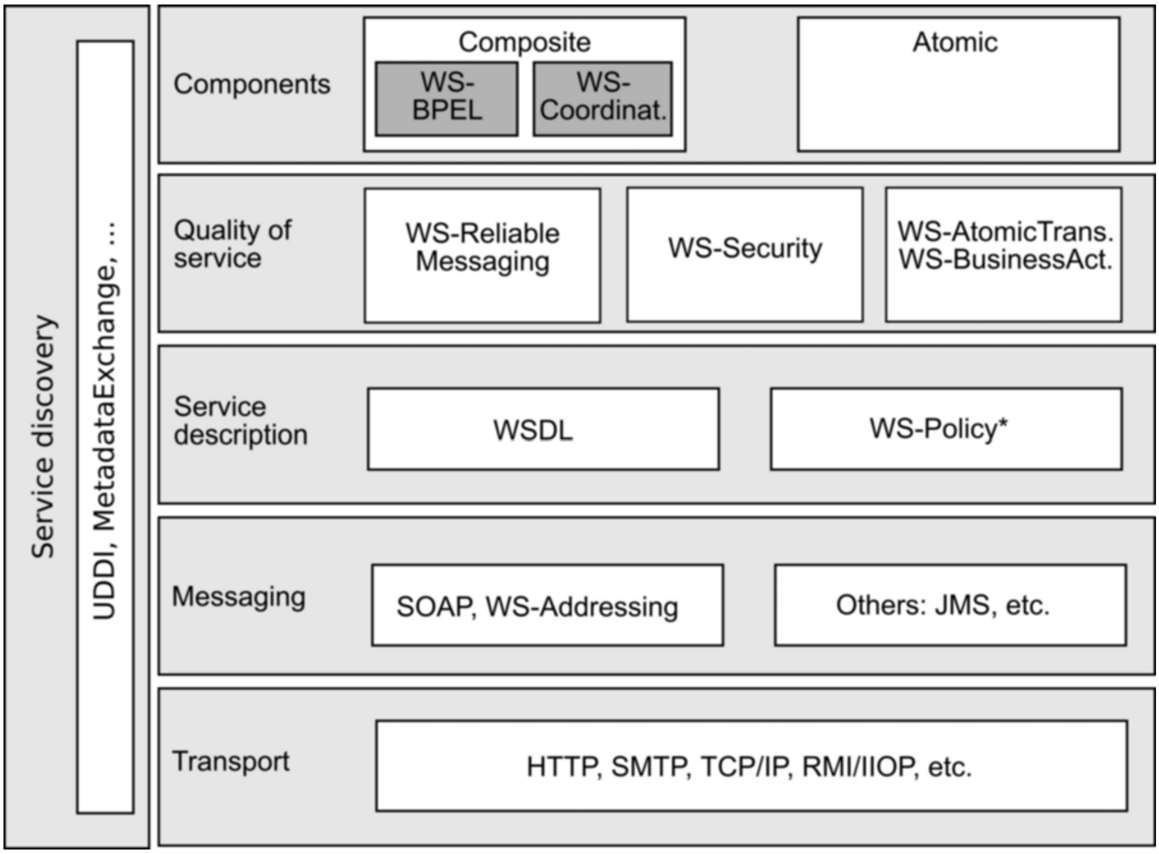

Figure 1: Layered Web Service Architecture

of "plug-compatible" software components that allow reducing the costs of developing software systems while at the same time allows increasing their capabilities. Though the origin of SOAs cannot be found on Web Service technology, but on previous efforts on the development of distributed systems (from Remote Procedure Calls to CORBA – *Common Object Request Broker Architecture*), it is true that Web Services have contributed deeply to their success and wide implementation, and constitute a good technology to realise this vision. Section 2 will describe this vision further and will focus on how Web Services contribute to it.

With the emergence and wide deployment of Web Service-based applications in different contexts, new requirements arise.

These include how to represent and exchange Web Service metadata, how to ensure the reliability of services, how to deal with security aspects such as authentication and authorisation, how to compose services and coordinate their executions in complex interactions, how to manage and monitor them, how to represent and deal with their state in the case of stateful resources, etc. To address all these requirements, new specifications have been created on top of the basic set of specifications. They will be described in Section 4.

Many challenges still exist in the development of Web Service-based applications. These challenges include service discovery, selection, composition, negotiation, dynamic configuration, invocation, monitoring and recovery. A large amount of research is being devoted to studying them and proposing solutions. Current research is focused on isolated specific aspects (for instance, improving service discovery algorithms using a combination of keywords, input and output datatype similarity, etc.) or on more comprehensive approaches, such as the ones proposed in the context of Semantic Web Services, which aim at augmenting the descriptions of Web Services with formal metadata related to the application domain, the preconditions and postconditions for the service execution, etc. This last approach would permit finding solutions to most of the aforementioned challenges, although with the additional cost of requiring a higher effort for service providers who have to annotate their services.

Other distributed and non-distributed technologies are also adopting Web Services as their means to implement their functionality: this is the case of datasets, which are being made available by Web Services, or of Grid Technologies, which are also adopting (and extending) Web Service specifications and components. All these research challenges will be described in Section 5.

Finally, in Section 6 we will be giving examples of applications based on Web Service technology. Applications range several domains: public administration, financial services, B2B (*Business to Business*) integration, enterprise application integration, etc.

2 Architecture

Web Service technology is not monolithic. It is defined by means of a set of specifications, each of which is focused on solving different needs of service oriented architectures. Even if the main Information Technology companies have different objectives in mind with respect to Web Service technology, they collaborate in the standardisation of these specifications. As a result, they have converged into a set of widely-adopted basic specifications that have contributed to the success of this approach to application development.

Figure 1, obtained from [1], presents a commonly agreed view of Web Service Architecture, structured along a set of functional layers with pointers to the most relevant specifications.

Now we summarise the role of each layer in the global architecture, and the main characteristics of the corresponding specifications:

- Web Services define a message-based architecture. The transport layer is focused on the set of protocols used to deliver these messages. Web Service specifications allow using any protocol in this layer, though most of them rely on HTTP and HTTPS.
- The Messaging layer defines protocols that specify the format of the exchanged messages, their source and target, the systems that can process them, etc. This layer is the core of the architecture and the two main specifications are SOAP and WS-Addressing.
- The Service description layer is used to express metadata about service capabilities, both from the functional and non-functional points of view. The most important specification in this layer is WSDL.
- The Service discovery layer allows users (either humans or systems) to look for services that can accomplish their goals, according to the metadata stored for them. The most important specification in this layer is UDDI.
- The Quality of Service layer is in charge of non-functional aspects of Web Services, which are relevant to make it possible to interact with them, such as reliable messaging, security, transaction management, etc.
- The Components layer is in charge of the composition of new services from existing services, and of the coordination of interacting services.

3 Basic Web Service Specifications: SOAP, WSDL and UDDI

3.1 SOAP

SOAP is an XML-based protocol used to define the messages to be exchanged in an heterogeneous and decentralised environment, independently of the transport protocol used, the type of communication established, and the rules for the interpretation of messages.

SOAP has been standardised by the W3C consortium [2], [3], and widely accepted by industry.

3.2 WSDL (Web Service Description Language)

WSDL is a language used to describe the public interface of a service, including its public functions, the service location and the way to access it. The aspects related to service invocation, operations and messages are described in an abstract way so that they can be linked later to a network protocol and to a specific message format [4].

WSDL has been also standardised by the W3C consortium, with different versions [5], [6].

3.3 UDDI (Universal Description Discovery and Integration)

UDDI [7] is used to describe an XML-based universal registry that stores different types of information about the Web Services in a system. The UDDI specification includes a set of APIs for the discovery and publication of services in repositories.

4 Other Specifications

Besides the previous three basic specifications, now we describe others that are in the standardisation process and define other aspects related to Service-Oriented Architectures.

4.1 Addressing: WS-Addressing

WS-Addressing allows identifying nodes that exchange messages, independently of the protocol used to transport them, by means of *endpoint references* (EPRs). This specification is currenlty under the standardisation process by the W3C, and some documents are already proposed as W3C Recommendations [8][9][10].

4.2 Policies: WS-Policy

WS-Policy defines a general framework to describe and combine, in an abstract way, different types of policies regarding the service access or execution features of the services in different domains. This includes aspects such as security, reliable messaging, transactions, etc. The information provided with WS-Policy complements the WSDL functional descriptions. As with the previous one, this specification is currently under standardisation by the W3C, with some working drafts [11][12].

4.3 Metadata Exchange: WS-MetadataExchange

WS-MetadataExchange [13] defines protocols to allow endpoint references to exchange their metadata. Metadata can include policies, WSDL descriptions, XML Schema datatype descriptions, etc.

4.4 Security: The WS-Security Family

Secure transport protocols like HTTPS are not enough for building secure service-oriented applications, since they usually imply the use of multi-step messages, participation of more than two entities, etc. The WS-Security family of specifications (which includes WS-Trust, WS-SecureConversation and WS-Federation, among others) proposes an interoperable way to combine existing security techniques in an interoperable way. Some of these specifications are OASIS (*Organization for the Advanced Struc-*

tured Information Standards) standards [14] and others are in the standardisation process.

4.5 Reliable Messaging: WS-Reliability and WS-ReliableMessaging

Independently of whether the lower-level protocols are realiable or not, other additional mechanisms have to be used to guarantee a reliable end-to-end message exchange. WS-ReliableMessaging [15] specifies three basic semantics that can be combined: ordered delivery, delivery of each message at least once and deliver of each message at most one. WS-Reliability [16] is similar to WS-ReliableMessaging and has been standardised by OASIS.

4.6 Transactions: The WS-Coordination Family

In complex interactions with multiple message exchanges it is difficult to guarantee that the final result of the interaction is coherent, due to the occurrence of errors, unexpected situations, etc. For this reason protocols for the coordination of transactions are being proposed. WS-Coordination specifies a general framework to define such coordination protocols. Among these protocols we can cite WS-AtomicTransaction, for applications that require the use of the classical model for atomic transactions ACID (*Atomicity, Consistency, Isolation and Durability*), and WS-BusinessActivity, for long-lasting business interactions where the ACID model is not adequate. These specifications are available at [17][18][19].

4.7 Composition: WS-BPEL

WS-BPEL (*Business Process Execution Language*, previously known as BPEL4WS) is a high-level language for defining service composition (aka business processes). Service composition descriptions can be executed and managed automatically by execution engines that are compatible with the language. The current BPEL4WS 1.1 [20] was sent to OASIS for standardisation and will be transformed into the WS-BPEL 2.0 specification.

4.8 State: WSRF

The Web Service interfaces aforementioned do not specify how service providers and requesters have to deal with the access to the resources they are wrapping when these resources are stateful. The WS-Resource [23] construct was proposed as a means to expressing this relationship, and the *WS-Resource framework* (WSRF [24]) is a set of Web Service specifications that define a rendering of the WS-Resource approach in terms of specific message exchanges and related XML definitions: *WS-ResourceProperties* (WSRF-RP), *WS-ResourceLifetime* (WSRF-RL), *WS-ServiceGroup* (WSRF-SG), and *WS-BaseFaults* (WSRF-BF). The aim of this specification is to allow programmers to declare and implement associations between a Web Service and one or more stateful resources, describing how the resource state is accessible through the Web Service interface and defining related mechanisms concerned with WS-Resource grouping and addressing.

The initial work on WSRF was performed by Globus Alliance and IBM (WSRF was first released in January 2004), as an initial refactoring of the concepts and interfaces developed in the OGSI V1.0 specification [25]. Now it is in the process of standardisation by OASIS[1].

5 Semantic Web Services

Current Web Service technologies (SOAP, WSDL and UDDI) operate at a syntactic level. Hence they normally require humans to search for appropriate Web Services to be used in an application and to combine them so that they fulfil the application objectives. Semantic Web Services are aimed at overcoming this human dependency by providing (semi)-automatic means to discover, select, and compose Web Services for an application, giving support as well to service mediation, execution and monitoring. The basis for providing such functionalities is the annotation (also known as *markup*) of Web Services with machine-understandable content that describes their preconditions and postconditions, their inputs and outputs, etc.

There are currently three leading approaches for the development of Semantic Web Services: WSDL-S, WSMO (*Web Service Modeling Ontology*) and OWL-S (*Ontology Web Language-S*). A comparison can be found in [21]. All of them are similar with respect to their design principles and to the types of problems that they aim to solve. However, they differ in the formal languages and models used to describe services, and in how they integrate with the underlying Web Service technology.

WSDL-S [26] uses the extensibility elements of WSDL in order to include semantic descriptions of Web Services. Extension attributes are used to associate WSDL entities with concepts in a semantic model, to handle structural differences between the Web Service schema elements and the semantic concepts, to specify preconditions and effects of each Web Service operation, and to specify a semantic category of the service, which can be seen as an extension to the UDDI registry information. The WSDL-S approach does not make any assumption about the formal language to be used to specify the semantics of Web Services. Hence it can be seen as complementary to the other two approaches.

WSMO [27] and OWL-S [28] propose ontology-based models for describing Web Services. The WSMO model is specified in the WSML (*Web Services Meta Language*) and can be dealt with by execution platforms like WSMX or IRS-III. WSMO descriptions specify preconditions, postconditions, assumptions and effects of Web Service operations, as well as non-functional properties of the service. The OWL-S model is specified in the OWL language and can be dealt with by the OWL-S virtual machine. Services are described according to their profile (what the service does), model (how to use the service and what happens when the service is used) and grounding (details on how

---

[1] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrf

requesters can access the service).

These approaches are under standardisation by different standardisation committees (W3C, OASIS). Open research issues in this work are still on the aspects related to automatic discovery, selection and composition of Web Services. There are also contests open for participation for these open issues[2].

6 Applications

The emergence of Service Oriented Architectures in organisations has spread the use of Web Services as the candidate technology to implement the design principles behind this paradigm. Below we describe the main application areas of Web Services.

6.1 Enterprise Application Integration (EAI)

Web Services have changed the way in which enterprises integrate their legacy applications, departments and information systems. Web Service standardisation is key to make it possible to share information and services with other partners, providers and customers, reducing the software development and maintenance costs and taking advantage of all the development and investments done in the enterprise.

Besides, the importance of Web Services in application integration has grown because of the popularity of *Enterprise Service Buses* (ESBs).

6.2 Business Process Management (BPM)

BPM solutions are linked to the maintenance of the lifecycle of business process, together with their automatisation, optimisation and design. Web Services constitute a key technology to achieve an open integration between inter and intra-organisational processes.

Specifications like BPEL have boosted the support for the orchestration of collaborative Web Service based processes.

6.3 Business-to-Business and E-Commerce (B2B and EC)

Collaboration in B2B requires an open and normalised exchange of information between organisations, so that they can converge into a common model and ensure the extension of their business and organisational boundaries to other organisations.

With respect to Electronic Commerce, Web Services are used in portals like Amazon.com, Yahoo.com, or eBay, using APIs based on REST. In the academic environment, analyses are being made about the impact of Web Services in virtual markets, contracts and e-provisioning.

6.4 Public Sector

Web Services are being used in all types of public administrations to offer services to citizens (change of address, tax payment, electronic notifications, certificates, etc.), and in those scenarios where the collaboration between different public administrations is required.

As an example, in the Health sector we can highlight the wide use of Web Services to make it possible for hospital information systems and other public Health-related services to interoperate. Similarly, Web Services are being used in other areas such as tourism, finance and insurances.

6.5 New Business Models

Web Services are being also considered as a gateway to new business models, since they provide an inter-connection of services that was not provided before. One example is the emergence of the concept of "*Software as a Service*" (SaaS) [22], which represents a new way to distribute software.

Translated by ***Oscar Corcho-García***

References

[1] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson. 2005. Web Services Platform Architecture, Prentice Hall, 2005. ISBN: 0-131-48874-0.

[2] Simple Object Access Protocol (SOAP) 1.1. W3C Note. May 08, 2000, <http://www.w3.org/TR/SOAP>.

[3] From SOAP/1.1 to SOAP Version 1.2 in 9 points. <http://www.w3.org/2003/06/soap11-soap12.html>.

[4] Web Services Description Language. Wikipedia. <http://en.wikipedia.org/wiki/Web_Services_Description_Language>.

[5] Web Services Description Language (WSDL) 1.1. W3C Note. March 15, 2001, <http://www.w3.org/TR/wsdl>.

[6] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. March 27, 2006, <http://www.w3.org/TR/wsdl20/>.

[7] Universal Description, Discovery and Integration of Business for the Web (UDDI), August, 2001 <http://www.uddi.org>.

[8] Web Services Addressing 1.0 – Core, W3C Recommendation, May 9, 2006. <http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/>.

[9] Web Services Addressing 1.0 - SOAP Binding, W3C Recommendation, May 9, 2006. <http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/>.

[10] Web Services Addressing 1.0 - WSDL Binding. W3C Candidate Recommendation , May 29, 2006. <http://www.w3.org/TR/2006/CR-ws-addr-wsdl-20060529/>.

[11] Web Services Policy 1.5 – Framework. W3C Working Draft. July 31, 2006. <http://www.w3.org/TR/2006/WD-ws-policy-20060731/>.

[12] Web Services Policy 1.5 – Attachment. W3C Working Draft. July 31, 2006. <http://www.w3.org/TR/2006/WD-ws-policy-attach-20060731/>.

[13] Web Services Metadata Exchange Specification (WS-MetadataExchange). September, 2004. <http://www-128.ibm.com/developerworks/library/specification/

[2] http://deri.stanford.edu/challenge/2006/

ws-mex/>.

[14] WS-Security Core Specification 1.1. OASIS Standard Specification. February 1, 2006. >http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>.

[15] Web Services Reliable Messaging Protocol (WS-ReliableMessaging). February, 2006. <http://www-128.ibm.com/developerworks/library/specification/ws-rm/>.

[16] Web Services Reliable Messaging TC WS-Reliability 1.1. OASIS Standard. November 15, 2004. <http://docs.oasis-open.org/wsrm/ws-reliability/v1.1/wsrm-ws_reliability-1.1-spec-os.pdf>.

[17] Web Services Coordination (WS-Coordination). November, 2004. <ftp://www6.software.ibm.com/software/developer/library/WS-Coordination.pdf>.

[18] Web Services Atomic Transaction (WS-Atomic Transaction). November, 2004. <ftp://www6.software.ibm.com/software/developer/library/WSAtomicTransaction.pdf>.

[19] Web Services Business Activity Framework (WS-BusinessActivity). November, 2004. <ftp://www6.software.ibm.com/software/developer/library/WS-BusinessActivity.pdf>.

[20] Business Process Execution Language for Web Services. Version 1.1 Specification. May, 2003. Available at <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel>.

[21] A. Polleres, R. Lara D4.2v0.1 Formal Mapping and Tool to OWL-S. Working Draft, December 2004 <http://www.wsmo.org/2004/d4/d4.2/v0.1/20041217/>.

[22] Software as a Service. Wikipedia. <http://en.wikipedia.org/wiki/Software_as_a_Service>.

[23] Modeling Stateful Resources with Web Services. <http://www-128.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf>.

[24] WSRF-Primer. <http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf>.

[25] K. Czajkowski et al. From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring and Evolution. <http://www.globus.org/wsrf/specs/ogsi_to_wsrf_1.0.pdf>.

[26] Web Service Semantics - WSDL-S. W3C Member Submission. November 7, 2005. <http://www.w3.org/Submission/WSDL-S/>.

[27] Web Service Modelling Ontology. <http://www.wsmo.org/>.

[28] DAML Services. <http://www.daml.org/services/owl-s/>.

